U. S. DEPARTMENT OF THE INTERIOR

U. S. GEOLOGICAL SURVEY

GPRMODV2: One-Dimensional Full Waveform Forward Modeling of Dispersive Ground Penetrating Radar Data Version 2.0

by

Michael H. Powers[1] and Gary R. Olhoeft[1,2]

Open File Report 95-58

January 1995

The U.S. Geological Survey preliminary computer program for modeling ground penetrating radar data is written in C for compilation with Intel 386/486 C Code Builder Kit, version 1.1a, to run under MS-DOS 3.0 or later on 80386/80387, 80486, or Pentium computers with 4 Mbyte or greater memory available to the program. Source code is included.

[1] U.S. Geological Survey, Box 25046, MS 964, Denver, CO 80225-0046
[2] Now with: Department of Geophysics, Colorado School of Mines, Golden, CO, 80401-1887

Contents

System Requirements

Hardware: This program will run on most standard IBM-compatible personal computers with an 80386/80387, 80486, or Pentium central processing unit. It requires 1) a display and graphics adapter that will support the VESA (1991) standard 1024x768x256 SVGA mode, 2) at least 4 Mbyte of memory available to the executable program, and 3) storage capacity (eg. hard disk) of at least 3 Mbyte to hold the required software.

Software: The minimum software to run this program includes MS-DOS 3.0 or higher operating system, the executable file, and at least one transmission wavelet file. The format of the transmission wavelet file is described in detail in this report. Other files containing radar field data can be used by the program, and are also described in this report. To change the source code and re-compile the program requires the source code, a program editor, a 32-bit C compiler, Olhoeft's (1994a) graphics library and include file, a 32-bit version of Pinson's (1991) window library and include file, standard C include files, and a C linker, librarian and binder. The program is only known to properly compile and link with the Intel 386/486 C Code Builder Kit, version 1.1a. Other compiler kits may or may not work.

Introduction

This report is meant to be a user's manual for the program. It contains the mechanics of which buttons to press, as well as descriptions of what happens mathematically after they are pressed. The format includes a brief section on the essential information for getting started immediately, a detailed description of the modeling algorithm, and full explanations of each utility module. The summary reiterates the purpose of the program by listing the required input and the expected output.

The program is designed to help the user obtain detailed information on the subsurface electromagnetic properties of a horizontally layered media by full waveform forward modeling of ground penetrating radar (GPR) data. Field radar data can be compared to a computer-generated model response to better understand the earth electrical properties at a field site. The user must start with a data file containing the antenna output wavelet (transmission wavelet). An earth model of up to fifteen layers must then be described, defining the complex relative dielectric permittivity and the complex relative magnetic permeability as functions of frequency, and the conductivity and thickness for each layer. The program assumptions include zero source-receiver offset; normal incidence; horizontal, layered, homogeneous, isotropic media; and no 3D, electric or magnetic polarization, near-field, or scattering (volume or rough surface) loss effects as the original wavelet is convolved with the appropriate transmission and reflection operators. The model response is displayed on the screen and, optionally, overlain onto an imported field trace.

Most radar antennas are loaded by the surface earth material and change their output response with changing coupling and surface properties. This effect is often expressed as a stretching (broadening) of the antenna output wavelet in time. Therefore, the user can optionally stretch the starting transmission wavelet such that the resulting model response best matches the frequency content of the field data.

The program enables the user to interactively change the earth model parameters and watch the response become more or less similar to the field data. The user gains an understanding of how GPR data is affected by different layer thickness geometries and material parameters. Once the best graphical fit is obtained, the user has one possible description of the shallow, subsurface electrical and magnetic properties versus depth at the field site.

A Master's thesis by Duke (1990) contained the original algorithms that have evolved into this program. His work was pulled together into a single executable program that was distributed as GPRMODEL version 1.0 (Powers and others, 1992). This version differs from version 1.0 in both small and large ways among the various subroutines. The major difference is the new ability to express the permittivity and permeability as functions of frequency for any layer. This version is fully dispersive in that each frequency component of the starting pulse is allowed to attenuate and travel at its own, possibly unique, rate. The program also contains a new interface that includes a utility for displaying 2D GPR field data and interactively selecting a field trace for model comparison. The utility allows hyperbolic curve-matching for initial estimates of average permittivity and depth to specific targets. The field data viewing module can be very useful on its own.

Getting Started

This section is a guide to immediately using the program. The distributed software on the diskette includes a *readme.txt* file that contains detailed instructions on installation and other useful information. You should start by reading this file. An *install.bat* file will load the necessary software onto the hard disk (eg. drive c:). Insert the diskette (usually in drive **a:** or **b:**). Type **(drive1)install (drive1) (drive2)** where (drive1) is the floppy drive and (drive2) is the hard disk drive. For example, to copy from drive b: to drive c: type **b:install b: c:**. The install routine creates two root subdirectories called *\hershey* and *\gprmod*. A font file is loaded into the first, and the necessary program files are loaded into the second. The install batch program then automatically calls the modeling program. Future program runs are started by moving to the *\gprmod* directory and typing **gprmodv2**. Type **gprmodv2 ?** for command line options.

After paging through the introductory screens, the display shows the Main Menu (figure 1). Available choices are shown with a capitalized, highlighted "hot-key". Selections are made by scrolling the highlight through the options with the Up and Down Arrow keys, and pressing Enter when the desired choice is highlighted or by pressing the "hot-key" of an available choice. (The user may need to set the contrast and brightness on a monochrome monitor such that the highlighted keys are visible.)

A starting transmission wavelet file must be input to the program before modeling can begin. All choices on the Main Menu will become active after successfully importing the transmission wavelet. Do this by selecting the 'T' hot-key option. A standard starting wavelet file named *ricker.trn* is distributed with the program. Type the pathname to this file. When it is successfully found, a popup screen should appear asking for the center frequency of the wavelet. Enter any number from 1 to 2000, depending on the modeling objectives. This choice is similar to choosing an antenna center frequency for a GPR field survey. Another way to import a starting wavelet and begin modeling is to read a previously saved model file. This will immediately load all the program variables with the values they had when the model file was saved. Two example model files and their associated starting wavelet and field data files are included on the distribution diskette.

After successfully introducing a starting wavelet, all of the Main Menu options become available as shown by their highlighted 'hot-keys' (figure 2). Press Enter while the eXecute model option is selected, or press the 'X' hot-key to see the model response. A default model with one meter thickness of air between the antenna and a half-space of metal is shown (figure 3). The right side of the screen shows a schematic cross-section of the model, where the horizontal lines are layer interfaces with the depth to the interface in meters shown just right of the line. The Right and Left Arrow keys move the highlight among the parameters. The highlighted parameter value can be increased with the Home, Up Arrow, and Page Up keys, and decreased with the End, Down Arrow, and Page Down keys. (This is most simply done with the numeric keypad when the Num-Lock key is toggled off.) A new model response is calculated and displayed by pressing the Enter or the Space Bar. To add or delete layers, press Shift-Enter to go back to the Main Menu, and select the 'M' hot-key option. Press F1 at any time for help and a summary of active keys, or consult the other sections of this report for further information.

```
GPR Model v2.0 by M.H.Powers & G.R.Olhoeft, USGS - Denver     F1=HELP
                              MAIN MENU
                  Transmission wavelet selection
                  field data display and import
                  display parameters and options
                  coupling ratio
                  gain of model and field scan
                  Read model file
                  save model file
                  model build/edit
                  heading of plot
                  Introductory screens
                  Quit

                              MESSAGES
 Read a file containing a wavelet that will be used as the source input
 to the earth model (transmission wavelet).  Different antennas have
 different source wavelets.  Cr will be reset to 1.0.
```

Figure 1. A snapshot of the screen showing the Main Menu upon entering the program. A transmitted starting wavelet must be imported before modeling can begin, or a pre-existing model file must be read into the program.

```
GPR Model v2.0 by M.H.Powers & G.R.Olhoeft, USGS - Denver     F1=HELP
                              MAIN MENU
         eXecute model
         Transmission wavelet selection
         Field data display and import
         Display parameters and options
         Coupling ratio
         Gain of model and field scan
         Read model file
         Save model file
         Model build/edit
         Heading of plot
         Introductory screens
         Quit
                              MESSAGES
 Calculate and display model response using current parameters.  The model
 response display allows interactive parameter changes.
```

Figure 2. A snapshot of the screen showing the Main Menu after importing a transmitted starting wavelet. Note that all options are now active as shown by their enhanced capital letters which are used as "hotkeys" for instant selection.

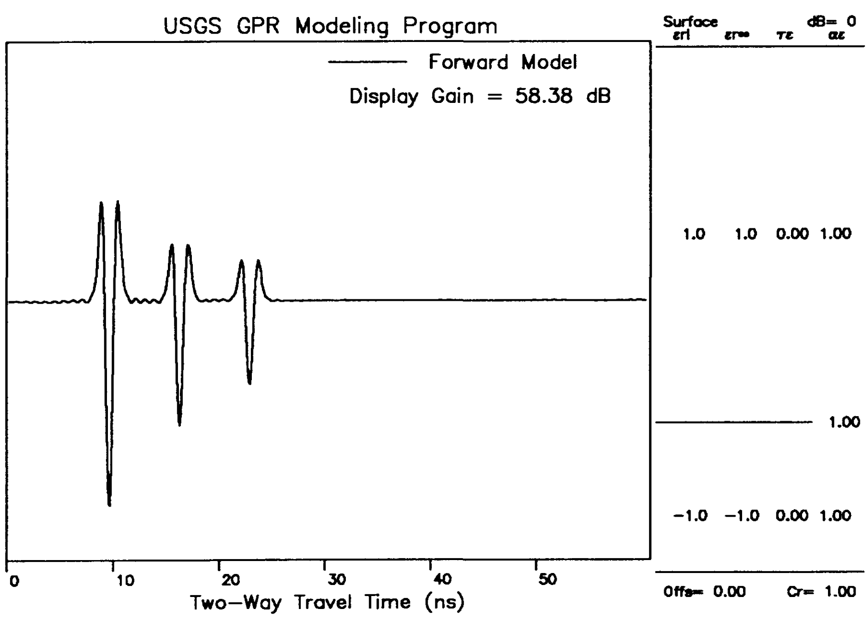

Figure 3. A hardcopy plot showing the default model response. The program starts by modeling the response of the starting wavelet after it has reflected from a layer of metal one meter away through air. As shown here, the primary and secondary multiple reflections are clearly visible. The right side of the screen shows a schematic cross-section of the model in depth. The Display Gain is automatically calculated to scale the response to match the screen. No range-gain (time-variable gain) has been applied in this plot.

Execute Model Algorithm

The purpose of this section is to give an understanding of the algorithm used to calculate a radar model response. This section is not meant to be a comprehensive discussion of electromagnetic theory, which can be found in Stratton (1940), Lorrain and Corson (1970), Ward and Hohmann (1988), and Balanis (1989). After the overview, the structure of this section follows the flow of the execution algorithm. Discussions of the theory for each process are presented as necessary to allow a complete understanding of the final displayed response.

Overview

The program calculates a one-dimensional response in the sense that all subsurface model layers are assumed to be flat and horizontal, and the source-receiver offset is zero. The only raypath considered encounters every interface with normal incidence (perpendicular), so the downward and upward legs of the raypath are identical, passing through the same space (i.e. lateral variations in layer properties are irrelevant). Schematically, the model response is the result of propagating a known starting wavelet through one centimeter of air beneath the antenna into the first subsurface layer. As it propagates downward it experiences geometric spreading, exponential material losses, and transmission effects at interface boundaries. The reflected energy moves upward and again experiences spreading, exponential material losses, and transmission effects. The upward-moving energy that arrives back at the antenna, transmitting from the first subsurface layer into the air as its final interface crossing, is displayed as the final result. All of the effects can be frequency dependent, and all except geometric spreading can be complex. An immediate inaccuracy is created by making a plane wave assumption from the instant the wavelet leaves the antenna. In reality there exists a near-field zone where boundaries do not reflect and transmit as calculated in this program. The user should understand that reflections from layer boundaries in this zone are unrealistic. The 1D nature of the program also leads to the exclusion of any electric or magnetic polarization effects. The model response is idealized to represent conditions where the source and receiver antennas have the same polarization, and none of the energy returned from any of the layers is lost through transmission or reflection polarization changes in the ground.

In the algorithm, the starting wavelet (possibly stretched or squeezed by the coupling ratio) is transformed into the frequency domain using a standard Fast Fourier Transform (FFT). For each frequency with non-zero amplitude, the material properties of each layer, the attenuation and propagation parameters, the reflection and transmission coefficients, and the sum effects of the raypath, including spreading, and (optionally) its multiples are respectively calculated and stored. When this process has been completed for the full frequency spectrum of the wavelet, the results are summed into one frequency-domain response which is transformed back to the time domain. Each of the steps is described in more detail below.

Cole-Cole Distribution

The material properties of each layer consist of thickness, conductivity, complex relative magnetic permeability, and complex relative dielectric permittivity. Relative permittivity and permeability (ε_r and μ_r) mean they are relative to the free space values as follows:

$$\varepsilon_r = \frac{\varepsilon}{\varepsilon_0} \qquad \text{and} \qquad \mu_r = \frac{\mu}{\mu_0},$$

where $\varepsilon_0 = 8.854 \times 10^{-12}$ Farads/meter and $\mu_0 = 4\pi \times 10^{-7}$ Henrys/meter, and ε and μ are the true values of permittivity and permeability in Farads/meter and Henrys/meter, respectively. The relative values are dimensionless.

Making μ_r and ε_r complex and frequency dependent requires that the user input complex functions of frequency for both of these parameters for every layer in the model. To make this simple and realistic according to laboratory analysis of many soil types (Olhoeft and Capron, 1994), the program uses Cole-Cole functions. The formulas are

$$\varepsilon_r' - i\varepsilon_r'' = \varepsilon_\infty + \frac{(\varepsilon_l - \varepsilon_\infty)}{(1 + (i\omega\tau_\varepsilon)^{\alpha_\varepsilon})}, \text{ and} \qquad \mu_r' - i\mu_r'' = \mu_\infty + \frac{(\mu_l - \mu_\infty)}{(1 + (i\omega\tau_\mu)^{\alpha_\mu})},$$

where ε_r', ε_r'', μ_r', and μ_r'' are the real and imaginary parts of the relative, complex permittivity and permeability, respectively; ε_∞, μ_∞, and ε_l and μ_l are the high and low frequency limits of permittivity and permeability, respectively; τ_ε and τ_μ are time constants of relaxation, α_ε and α_μ are Cole-Cole time constant distribution parameters (Cole and Cole, 1941), ω is angular frequency ($2\pi f$ where f is cycles per second), and i is the square root of negative one. In each layer, for both permittivity and permeability, four values are given (low frequency limit, high frequency limit, time constant, and distribution parameter). If the high and low limits are equal to each other, the property value is real and independent of frequency (and the time constant and the distribution parameter are irrelevant). When the high and low frequency limits are different, the functions have a simple, smooth decrease with frequency as shown in figure 4. The time constant of relaxation, τ_ε or τ_μ, controls where in frequency the decrease from the low frequency value to the high frequency value takes place. The slope on the step is controlled by the distribution parameter, α_ε or α_μ, which varies from 1.0 to 0.0 for steepest to smoothest slope. When either α_ε or α_μ is zero the corresponding permittivity or permeability is a constant value midway between the high and low frequency limits.

Physically, the dielectric permittivity of a material is its ability to support charge separation (polarization) phenomena. These generally include electronic, molecular, ionic, orientational, and interfacial polarization processes, which are ordered here from fastest to slowest with regard to keeping up (staying in phase) with an alternating electric field. As the frequency increases and some of the polarization processes cannot keep up

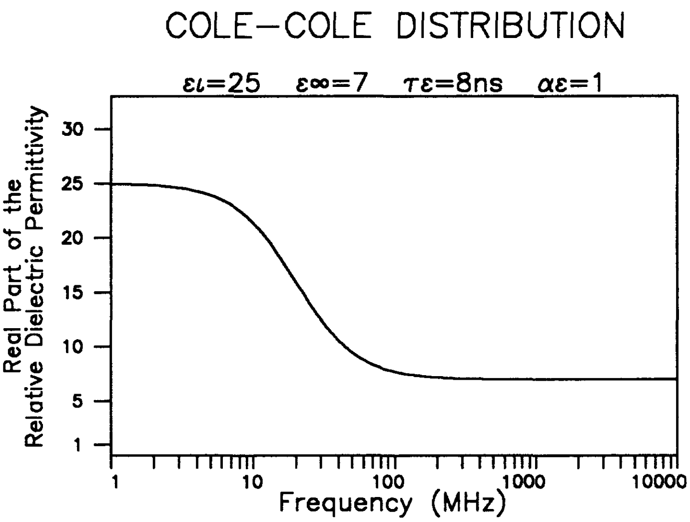

Figure 4. A plot of the real part of the complex relative dielectric permittivity versus frequency as determined by the Cole-Cole function. The Cole-Cole parameters for this plot are as shown. The user can interactively vary these four parameters for any layer. This plot is also available interactively as an aid in understanding the model effects on the response.

(stay in phase), the dielectric permittivity decreases. For particular materials, certain processes dominate such that the greatest rate of change of permittivity with frequency occurs when the dominant polarization processes start to lose pace (fall out of phase) with the flipping electric field. It is this process that the Cole-Cole function is meant to describe. See Bottcher and Bordewijk (1978) or Olhoeft (1981) for more information on this subject. The magnetic permeability for most earth materials is less likely to exhibit strong relaxation phenomena effects, but when they exist (usually due to the presence of iron) they can be described similarly with the Cole-Cole function (Olhoeft and Strangway, 1974; Olhoeft and Capron, 1994).

Attenuation and Propagation Parameters Alpha and Beta

The frequency domain representation of the starting wave equation for the program is

$$\frac{\partial^2 \mathbf{E}}{\partial z^2} = -k^2\mathbf{E}(z) \qquad \text{or} \qquad \frac{\partial^2 \mathbf{E}}{\partial z^2} = \gamma^2\mathbf{E}(z),$$

where **E** is the electric field vector, k is the wavenumber (or magnitude of the wave vector in the direction of propagation), γ is known as the propagation vector, and z is distance along the direction of propagation. The wave vector and propagation vector are related such that

$$\gamma = ik \quad \text{and} \quad k = (\beta - i\alpha) \quad \text{or} \quad \gamma = (\alpha + i\beta),$$

where α and β, the real and imaginary parts of the propagation vector, or the imaginary and real parts of the wave vector, respectively, are more fully defined below. For electromagnetic propagation through lossy material, the wavenumber, *k*, is determined by Maxwell's equations and expressed here as

$$k^2 = \omega^2(\mu_r' - i\mu_r'')(\varepsilon_r' - i\varepsilon_r'') + i\omega(\mu_r' - i\mu_r'')\sigma \quad = [\omega(\mu_r' - i\mu_r'')][\omega(\varepsilon_r' - i\varepsilon_r'') + i\sigma)],$$

where σ is the real-valued conductivity, and the other symbols are as defined in the previous sub-section.

The terms α and β are defined as the attenuation and phase constants, respectively (magnitudes of the attenuation and phase vectors in the direction of propagation) according to the IEEE convention (IEEE Standard 211, 1990), and can be found by separating the above equation into real and imaginary parts and solving simultaneously. They are

$$\alpha = \left(\frac{\omega}{c}\right)\sqrt{\left[\frac{\sqrt{A^2+B^2}-A}{2}\right]} \quad \text{and} \quad \beta = \left(\frac{\omega}{c}\right)\sqrt{\left[\frac{\sqrt{A^2+B^2}+A}{2}\right]},$$

where c is the speed of light in free space, and

$$A = \mu_r'\varepsilon_r' - \mu_r''(\varepsilon_r'' + \frac{\sigma}{\omega\varepsilon_0}) \quad \text{and} \quad B = \mu_r''\varepsilon_r' + \mu_r'(\varepsilon_r'' + \frac{\sigma}{\omega\varepsilon_0}).$$

For a given frequency, after the Cole-Cole function is used to determine the real and imaginary values of the relative permittivity and permeability (ε_r', ε_r'', μ_r', and μ_r''), α and β are calculated for every layer according to the above formulas. The values of α and β are used to determine the complex propagation losses and reflection and transmission effects of the wavelet through the model, as described in the next section.

The loss tangent in this version of the program is the total electromagnetic loss tangent. (In the first version, the magnetic permeability was assumed to be a constant real value and all references to the loss tangent meant just the electrical loss tangent.) The total electromagnetic loss tangent is α over β, or the ratio of the imaginary over the real part of the wavenumber. Physically, this is the phase difference between the vectors **E** and **H** (Lorrain and Corson, 1970), defined as $\theta_{\mathbf{EH}}$. The vectors **E**, **J**, **H**, and **B** are the electric field, the electric current density (including displacement currents), the magnetic field, and the magnetic induction, respectively. When the electrical loss tangent includes conductive losses it can be defined (as in version 1.0) as

$$\tan\delta_e = \frac{\varepsilon_r'' + \frac{\sigma}{\omega\varepsilon_0}}{\varepsilon_r'} = \cot\theta_{\mathbf{EJ}},$$

where θ_{EJ} is the phase angle between **E** and **J**. The magnetic loss tangent is

$$\tan\delta_m = \frac{\mu_r''}{\mu_r'} = \cot\theta_{BH},$$

where θ_{BH} is the phase angle between **B** and **H**. The total electromagnetic loss tangent is

$$\tan\delta_{em} = \tan\left(\frac{\delta_e + \delta_m}{2}\right) = \frac{\alpha}{\beta} = \cot\theta_{EH}.$$

Although the value of the total electromagnetic loss tangent is not used directly in the calculation of any propagation effects, the frequency where it is at a minimum for a particular material is where the radar waves will propagate with the best combination of highest resolution and least loss through the material. Conversely, when $\tan\delta_{em}$ has a peak as a function of frequency, it identifies a frequency range of high dynamic energy loss. As another way of explaining the Cole-Cole parameters, τ_ε, τ_μ, α_ε and α_μ, the time constants determine the frequency where the peak occurs in the respective functions of $\tan\delta_e$ or $\tan\delta_m$ versus frequency, and the distribution parameters determine the slope angles away from the peak on a log-log plot of $\tan\delta_e$ or $\tan\delta_m$ versus frequency. The total electromagnetic loss tangent, $\tan\delta_{em}$, is the combination of losses as defined by the above equation.

Reflection, Transmission and Attenuation

The formulas for complex reflection and transmission coefficients are simple given flat, smooth interfaces, normal incidence, and no polarization or near-field effects. They are

$$\mathrm{RC} = \frac{(\mu_{r2}k_1 - \mu_{r1}k_2)}{(\mu_{r2}k_1 + \mu_{r1}k_2)} \quad \text{and} \quad \mathrm{TC} = \frac{2\mu_{r2}k_1}{(\mu_{r2}k_1 + \mu_{r1}k_2)},$$

where k_1 and k_2 are the complex wavenumbers (β_1-$i\alpha_1$ and β_2-$i\alpha_2$), and μ_{r1} and μ_{r2} are the complex, relative magnetic permeabilities of two adjacent layer materials. For a given frequency, the algorithm exploits some simple relationships to calculate and store four coefficients for each layer; RC(up), RC(down), TC(up), and TC(down). When layer one is above layer two, the above formulas refer to RC(up) and TC(down). At a given interface, the relationships between the four coefficients are

RC(down) = -RC(up),
TC(down) = 1 + RC(up), and
TC(up) = 1 - RC(up).

The attenuation due to two-way travel through a layer is also calculated and stored for each layer for a given frequency. The formula is

$$\mathbf{ATT}_i = e^{-2\alpha_i(z_{i+1}-z_i)},$$

where α_i is the imaginary part of the wavenumber and the expression $(z_{i+1}-z_i)$ is just the thickness of the layer. Now, for a given frequency in the set of all frequencies with non-zero amplitude within the starting wavelet spectrum, the effects of propagation through the model and reflection from each boundary can be calculated by multiplying the appropriate frequency domain terms. For example, when the air-surface boundary is considered the 0^{th} layer, reflection from the 1^{st} layer will include multiplying TC_0(down), $\mathbf{ATT}_1$, RC_1(up), and TC_0(up). However, the reflection event is not complete until it also includes the geometric spreading effects and a phase shift representing the correct traveltime delay.

Geometric Spreading

True antennas have radiation patterns that vary the amount of energy output as a function of angle away from the normal to the antenna. Furthermore, this pattern changes with the electromagnetic properties of the host material near the antenna. In this program the limiting assumption of normal incidence (zero-offset, horizontal layers) allows the radiation pattern to be ignored. The energy being referred to in this section is assumed to be that which leaves the center of the antenna in a direction normal to the plane of the antenna.

As the energy leaves the antenna and propagates into the earth, it spreads such that the initial energy output is expanded over an ever-increasing spherical surface. The amount of energy that hits a reflector at depth and returns to the antenna is only a small fraction of the energy that left the antenna. The shape of the spreading wavefront is controlled by the velocity profile of the host material, and is only truly spherical when the host material has a constant isotropic, homogeneous velocity.

This program uses an algorithm derived from that presented in May and Hron (1978), and from information given in Balanis (1989). It considers normal incidence at all interfaces. If A_0 represents the amplitude measured at a distance R_0 from the source, then the amplitude after considering only the spreading effects from two-way travel to the i^{th} layer can be expressed as A_0M_i where

$$M_i = \frac{R_0}{(R_0 + S_i)} \qquad \text{and,}$$

$$\mathbf{S}_i = 2\left[\mathbf{z}_0 + \left(\frac{\beta_0}{\beta_1}\right)(\mathbf{z}_1 - \mathbf{z}_0) + \left(\frac{\beta_1}{\beta_2}\right)(\mathbf{z}_2 - \mathbf{z}_1) + \ \dots \ + \left(\frac{\beta_{i-1}}{\beta_i}\right)(\mathbf{z}_i - \mathbf{z}_{i-1})\right].$$

The z values are layer thicknesses and the β terms are the real parts of the wavenumbers for each layer as given above. Since velocity is ω/β, and this spreading effect is computed for a given frequency within the loop over all frequencies, the above formula can be understood physically in terms of velocity ratios at the interfaces. (An increasing velocity with depth will lead to a faster spreading, and a decreasing velocity with depth will lead to a slower spreading than that for constant velocity.)

Phase Shift for Time Delay

Because dispersion involves changes in velocity and attenuation with frequency, the heart of dispersive modeling is the ability to compute a separate time delay for each frequency for the traveltime to a particular reflection event. The first version of this program (Powers and others, 1992) was not really dispersive in that it computed a complete, separate frequency spectrum response for each arrival event, transformed them all to separate time domain events, and then did a constant time shift for each event using a non-dispersive velocity for each layer. The time-shifted events were summed to create the final output.

The current algorithm calculates a separate velocity and corresponding time-delay for each frequency for each event. Because a time-shift in the time domain is just a phase-shift in the frequency domain according to the simple relation

$$f(t-T) = F(\omega)e^{i\omega T},$$

this just involves another multiplication in the frequency domain.

As described in the previous section, for a given frequency and a particular reflection event, the response is the product of the appropriate reflection and transmission coefficients, the layer attenuations, and the spreading effects. At this point it now also includes a phase-shift multiplication to the real and imaginary values according to Euler's formula

$$e^{i\omega T} = \cos(\omega T) + i\sin(\omega T),$$

where ω is the angular frequency and T is the appropriate (frequency-dependent) two-way traveltime delay for the given event.

Multiple Reflections

In a real survey, arrival energy can come from multiple paths as well as from primary reflections. (Note that in a real survey arrival energy also comes from full 3D phenomena with constructive and destructive interference contributing to create the final, recorded field trace (Olhoeft, 1994b). This must be kept in mind when trying to match the simple, limited model responses to field data.) In the default mode, this program includes up to 12 multiple arrivals. They can be switched off at any time.

Two limiting rules were created when deciding which multiples to compute. They

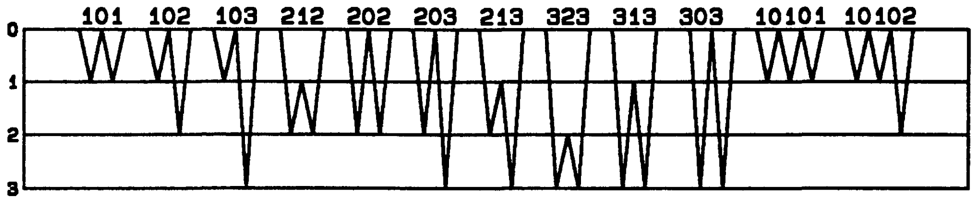

Figure 5. A diagram showing the twelve multiples optionally calculated by the program. Note that in the program every reflection and transmission is normal to every horizontal layer interface, as the source-receiver offset is always zero. Offsets are shown here only to clearly display the multiple reflection paths. A path is described by stating in sequence the interface numbers where a reflection occurs (up or down), and the labels describe the multiple paths. The interface numbers of the subsurface model are shown on the left, where the air-earth interface is number zero.

are 1) no multiples from any layer deeper than interface number 3 (the air-earth interface is considered number 0), and 2) only first multiples (three reflections) are included. This results in 10 unique multiple events. The second rule was then relaxed to include two particular secondary multiple events. When an event is described by stating the interface number where a reflection (either up or down) occurs, the primary reflection events are simply the numbers 1,2, etc. The first multiples are described by three numbers, such that the original ten are 101, 102, 202, 212, 103, 203, 303, 213, 313, and 323. The two exceptional secondary multiples are 10101 and 10102. Figure 5 shows all of these multiples in diagram form. Equivalents such as 102 and 201 or 10102 and 10201 need not be separately calculated, but they are taken into account by the program.

The amplitudes of any of these multiples, and hence their importance to the final model response, are always dependent on the model parameters. In most cases, the longer the path, the lower the resultant amplitude. It was this general effect that was used in setting the two rules described above. Obviously, models can be created where deeper or longer multiples would show up were they to be calculated. Unfortunately, in this program, and any other that considers specific event paths (ray-tracing methods), some limit must be placed on which events to include. In this program, the user can only set a single, interactive switch to include, or not include, all twelve of the above multiples.

Post-Computation Effects on the Model Response

After all of the primary and multiple events are computed as described in the preceding paragraphs for all relevant frequencies, the resultant response is transformed back into the time domain. Before being displayed it is time-shifted according to the parameter 'offset', gained according to the current user-determined (time-variable) range-gain function, and possibly scaled or clipped to fit the screen.

A positive time shift (offset) simply moves each value to a later time, substituting zeroes in the front (early time) and losing the values shifted off the back (late time). A negative time shift moves the values to earlier times, losing the previous early time response and replacing the late time with zeroes. The amount of time shift is determined by the user-controlled parameter 'offset'. Certain field data formats contain a parameter describing which sample number corresponds to zero time. When present, 'offset' is set by this parameter, but the user can always reset it.

After accounting for 'offset', the response is gained according to the current (time-variable) range-gain function. The range-gain function always starts at the first sample. If the response has been time shifted, then a particular time event has been moved to a new part of the range-gain function. This simulates the effect of common commercial field data acquisition systems. By default there is no range-gain for modeling without field data comparison, but the user can create a function using the module described later. When field data are imported for comparison, the range-gain is set to that of the field data. It can still be changed any time by the user.

In the case of modeling without field data comparison, the program determines a scalar that will set the maximum response amplitude to 80% of the screen display window maximum. Every value in the model response is multiplied by this single number. It is converted to decibels according to

decibel value = 20 * log10(scalar value),

and the decibel value is shown on the screen as the Display Gain. In the case of field data comparison, this step is skipped.

When a field data trace has been imported for comparison with the model response, the screen amplitude scale is set according to the number of bits per sample in the field data. With 8-bit data, the screen is set to vary from -127 to 128, for 16-bit data it varies from -32767 to 32768. This way the field data can be plotted directly with no scaling, and it will look as it did in the field. To approximately match the model response amplitudes to the field data, the starting wavelet is scaled before it is transformed to the frequency domain. The scalar is determined by removing any range-gain from the field trace, finding its maximum amplitude, and setting the maximum amplitude of the starting wavelet to the same value. Because the model response and the field trace always have the same (time-variable) range-gain applied, if any, their relative amplitudes will only differ by a single scalar. Hence, the user can apply a simple interactive display gain between ±96 dB to the model response only. The model response is scaled by the display gain value given in decibels and set by the user interactively. This is the parameter shown on the screen as 'db'. It is originally set to zero, but can be changed at any time. It is always used to scale the model response only, whether field data has been imported or not.

Finally, just before plotting the data, they are clipped such that amplitudes greater or less than the maximum and minimum, respectively, on the screen display are set equal to the maximum or minimum. This is required to avoid problems with the plot utility. After plotting the model response and the field data (if imported), the program flow moves to an interactive parameter input module, as described in the next section.

Interactive Options

Within the eXecute model selection, after the model response has been computed and displayed as described above, the right side of the screen shows a schematic cross-section of the current model. The ground surface is at the top, and horizontal white lines represent layer interfaces going down the screen with increasing depth. The number immediately to the right of each interface line is the depth to that interface in meters. The last layer is assumed to have infinite depth. Initially, there are four numbers shown within each layer. These are the Cole-Cole parameters used to describe the relative dielectric permittivity (dielectric constant) function within each layer, as described previously. If the first two numbers, representing the low and high frequency limits of permittivity, respectively, are equal, then the last two numbers are ignored and the permittivity for that layer is constant over all frequencies. If the first two numbers are set to -1, it is taken as a flag for metal, and the permittivity and the conductivity for that layer are taken to be very large numbers in the computations. The second two numbers are dielectric τ_ε and α_ε of the Cole-Cole parameters described previously.

The user can change any of the parameters shown on the right side of the screen by first highlighting the number to be changed. The Right and Left Arrows control the selection of the highlighted parameter. To change the value of the highlighted number, the numeric keypad is used. With Num-Lock OFF, the 7,8, and 9 keys are Home, Up Arrow, and Page Up, respectively. These are used to increase the value of the number by 1.0, 0.1, and 0.01, respectively. The 1,2, and 3 keys are End, Down Arrow, and Page Down, respectively. They decrease the value of the highlighted number by 1.0, 0.1, and 0.01, respectively. Since the 4 and 6 keys are Left and Right Arrow, respectively, with the Num-Lock OFF, the user can easily move among the parameters and change the values as desired. The model response is not recomputed until either Enter or Space Bar have been pressed. At that time the current values are used as shown on the screen.

To change the conductivity of any layer, the 'C' key replaces the display of the four Cole-Cole parameters within each layer with the single value of conductivity for the layer. Again, the user can use the Right and Left Arrows to move the highlight among the parameters shown, and increase or decrease the value of the highlighted parameter as described above. At any time the 'E' key will again display the relative dielectric permittivity parameters, making them available for change.

The 'M' key changes the displayed numbers to the relative magnetic permeability Cole-Cole function parameters. These control the function of magnetic permeability versus frequency in the same way as described above for the dielectric permittivity.

The 'M', 'C', and 'E' keys are active at any time to allow changes to any parameter, and the model response is not recomputed when they are pressed. The 'S' and 'R' keys do immediately recompute the model. The 'S' key is a switch for turning the twelve multiples on or off, and the 'R' key reverses the polarity of the starting wavelet.

The other interactive parameters include the depths to any layer interface. The minimum and maximum layer thicknesses are 1 centimeter and 1000 meters, respectively. Also, the three parameters 'offs', 'Cr', and 'db' can be highlighted and changed interactively. The response zero time is shifted right or left by the amount 'offs', which is given in nanoseconds. It is the value described previously as 'offset'. The coupling ratio is shown as 'Cr', and can vary from 5.0 to some minimum that is dependent on the sampling interval. A value of 1.0 means the center frequency of the starting wavelet is about equal to that originally defined by the values read from disk. Increasing the coupling ratio stretches the starting wavelet in time, as described in a later section, and decreases its center frequency. Decreasing the coupling ratio squeezes the starting wavelet in time, increasing its effective center frequency. The value of 'db' is given in decibels, and it is a constant multiplier applied to the model response only after all other computations are completed. (See the above section on Post Computation Effects on the Model Response for more information).

The only subsurface model description parameter that cannot be changed interactively is the number of layers. To add or delete one or more layers, the user must press Shift-Enter to return to the Main Menu, and press 'M' to enter the Model build/edit utility. This utility is described later in this document.

The 'F' key is also active. This key replaces the plot of the model response with plots of the power spectrum of the starting wavelet, and either the relative dielectric permittivity or the relative magnetic permeability along with the total electromagnetic loss tangent as functions of frequency. Within this frequency domain plot utility, the 'M', 'E', and 'C' keys still can be used to switch among the parameters, and all parameters can still be changed as before. Pressing Enter or Space Bar will now recompute the displayed functions. The 'L' key is also active as a switch between a log or linear frequency scale. The linear scale is over the frequencies included in the current starting wavelet. With this utility the user can better understand how the Cole-Cole parameters create the frequency-dependent functions, and can see if and when dispersion is important within the current frequency range. Shift-Enter is used to return to the eXecute model screen, where all changes to the material properties have been retained. There is no Escape option to leave this plot without saving the material property changes. All changes made here are equivalent to interactively changing parameter values on the primary eXecute model screen.

Within the frequency domain plot, the F2 key is active. Pressing it will open an automatic parameter estimation routine that is based on volume percentages of soil constituents. Active keys are uppercase and lowercase letters 'S', 'C', 'W', and 'I' used to increase (uppercase) or decrease (lowercase) the volume percent of sand, clay, water, and iron, respectively in a theoretical soil type. The soil mix is shown at the bottom of the screen. A fifth material, air, is included such that the sum of the five volume percentages always equals 100. If the volume percentage of air is zero, then none of the other four materials can be increased until one of them is decreased. Pressing Enter or Space Bar at any time will set the parameters of the current layer equal to a representation of the current soil mix. Shift-Enter can be used to exit this automatic parameter estimation utility and save the new parameters. Escape will exit without saving the new parameters.

The calculations of the dispersive dielectric permittivity and magnetic permeability Cole-Cole parameters, and the constant conductivity, for a given soil mixture are performed according to established mixing laws when possible. For dry soils, the mixing laws come from Olhoeft and Strangway (1975). When water is present, the Bruggeman-Hanai-Sen (BHS) formula is used (Sen and others, 1981). The iron mixing is based upon the experimental data in Olhoeft and Capron (1994). It is possible here to have up to a five-phase mixing problem, and the references consider only two-phase mixing. Considerable personal experience and subjective assumptions have been used in writing this utility. It should be used only as a possible starting estimate for the Cole-Cole parameters of a given soil mixture, and not as an absolute truth. To vary conductivity for varying salinity of water, directly change the conductivity as the mixing model does not explicitly include the effects of salinity.

Pressing 'F1' gives a help text screen at almost any time in the program. It can be used to give a succinct summary of the active keys during any interactive execution.

Model Display and Output Control

The 'D' hotkey on the Main Menu accesses a utility that allows the user to change the begin and end times on the model display. The entire model response is still calculated, and these parameters only control the display. They allow the user to zoom in on a particular arrival event if desired. The end time as set here is often not honored on the model response display. The reason for this has to do with time-domain wrap-around effects of phase shifting in the frequency domain. An event is zeroed if its arrival time puts it past the total time of the model (where that part would wrap around to the early time if not zeroed). The late time, where some frequencies within the pulse must be zeroed, is not allowed to be displayed because it would show a distorted response. To get around this, the user can increase the total time of the response by increasing the number of samples or the sample interval time. Note that neither of these can be changed when field data has been imported, as they are fixed by the field data record parameters.

Users with a model and a field data trace, who want to increase the total time of the model response while keeping the model, can enter the name of a field data file (under the 'F' key from the Main Menu) as UNTITLED in capital letters as shown. The field trace will no longer be displayed, but all other model parameters will be unchanged, and the number of samples or the sample interval time can now be changed.

The zero time shift known as 'offset' is also available for change within this utility, as the user may want to set the begin time to the offset time. To change the start time of the model response display shown on the eXecute model screen, press 'B' or use the arrows to move the cursor to the same line as the Begin time parameter and press Enter. After editing the time and pressing Enter, check the parameter value shown to ensure that it is as desired. The end time and offset time can be similarly edited.

The number of samples and the sample interval of the output model response are displayed here, along with the corresponding total time of the response. If no field data have been imported, then both 'S' and 'N' are active hot keys used to change the sample interval time and the number of samples, respectively.

The number of samples must be a power of two to facilitate the conversions between the time and frequency domains. It is up to the user to be aware of the effects of making the sample interval time too large or too small for particular frequencies. Keep in mind that the Nyquist frequency (highest frequency accurately distinguished from digital data) for a particular sample interval time, Δt, is equal to $1/(2 \cdot \Delta t)$. Also, remember that the frequency step size is equal to the reciprocal of the total response time. The total response time is shown on this screen as an aid to the user. It cannot be edited directly, as it is always the product of the current sample interval time and the number of samples.

Starting Transmission Wavelet

In the program, all modeling is the result of performing various effects on some initial signal. This signal must be digitized and stored on disk in the format described in this section. It is imported into the program prior to any modeling computations. Physically, it should represent the electric field energy entering the ground during a GPR survey.

This transmission wavelet is very hard to accurately understand in any real survey, because it is determined not only by the integration of the antenna and the system electronics, but also the interactions between the antenna and the ground surface. As the electrical properties of the near-field zone around the antenna change, whether from changing ground conditions or coupling changes (variations in the thickness of air between the antenna and the ground as the antenna bumps along), the output of the antenna changes accordingly. The biggest effect, and the only one compensated for in this program, is a ground-loading that shifts the output center frequency down as the ground permittivity goes up.

The module titled Coupling ratio on the Main Menu displays the currently loaded starting wavelet as a function of time and frequency (figure 6). Increasing the coupling ratio (Cr) will stretch the wavelet in time and shift the center frequency down. The coupling ratio parameter can also be varied interactively while executing the model response.

This program includes a single starting wavelet file called *ricker.trn*. It is a Ricker wavelet as defined by Sheriff (1984) that can be set for any center frequency between 1 and 2000 MHz. Other wavelets can be created by the user and imported if they are stored as a binary file of 513 values, each value 8 bytes long (double precision). The C code to read such a file into an array called `stwv[]` is as follows:

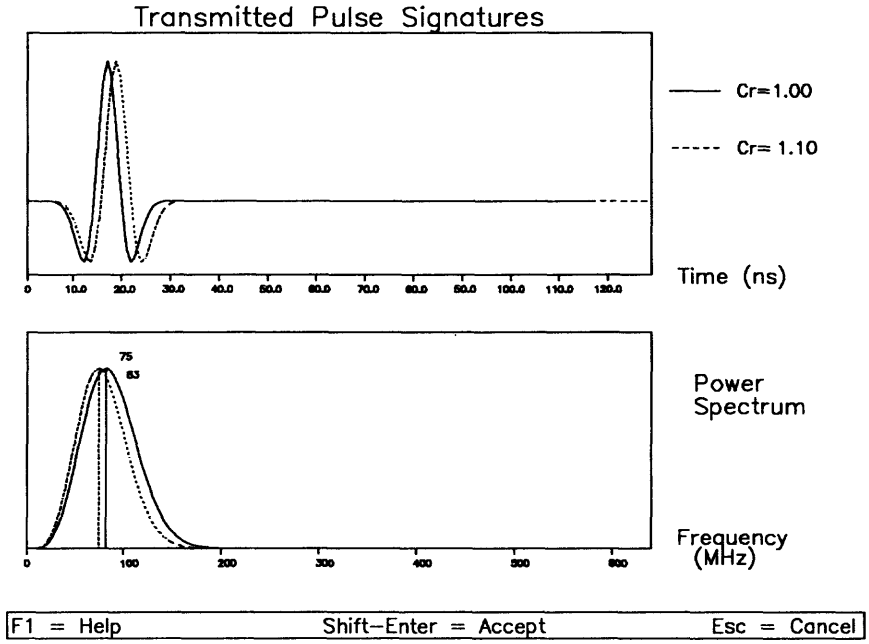

Figure 6. A hardcopy plot of the file created by pressing the 'P' key from the Coupling ratio module. An 80 MHz Ricker wavelet is shown.

```
char filename[80]="custom.trn";
double stwv[513], total_time, dt;
FILE *f_tran;

f_tran=fopen(filename,"rb");
fread(stwv,sizeof(double),513,f_tran);
fclose(f_tran);
dt=stwv[512];
total_time=511.*dt;
```

Note that the last (513th) value must be the sample interval time in nanoseconds.

The first version of this program (Powers and others, 1992) included three custom starting wavelets that represented our determinations of the output pulses in air from our antennas of 300, 500, and 900 MHz center frequency.

Field Data

This module allows the user to view field data as a 2D image file, and, optionally, to select a single trace for import to the modeling routine for comparison. **Note that this module needs at least 4 MB of available memory to display even the smallest field datasets.** The field data must be stored on disk in a recognizable format. Two commercial formats are supported: 1) Sensors & Software's *dt1* data format that always includes a separate header file with the extension *.hd* for every data file, and 2) a *.dzt* format from Geophysical Survey Systems, Inc. (GSSI) that is generated by the SIR10A+

system with version 4.2 software. The program will also read generic fixed-bit (8,16, or 32) datasets that can be described by the user in terms of bits per sample, samples per trace, bytes of header per file, and bytes of header per trace. In this latter case no range-gain information is available, and the user must know the total record time.

A utility program is available for converting data to the *.dzt* format after determining the range-gain and total record time (using other available utilities if necessary). It was specifically written to convert SIR7 generated, DT6000 tape recorded data, but it can be used to convert any known fixed-bit datasets. Descriptions of the supported commercial formats and the conversion utility programs are given in the appendices.

After accepting the name of a field data file, the program will attempt to open the file and recognize the format. Assuming this is successful, the user is shown a summary of information about the file, and given the options of proceeding to read the data or cancelling the request. The display module is designed to read in as much of the data as the computer's memory will hold. It then closes the file and allows the user to page the screen display forward and backward through the dataset (if it is larger than one screen width, which varies with the graphics mode). If the field dataset is larger than the available memory can hold, the user is prompted to specify a first and last trace number as limits on the amount to read into memory.

When the data appears on the screen, it is not necessarily an exact display of the stored values. The resolution of the gray scale is only 8-bit, so 16-bit datasets are converted down to 8-bit values using a simple, truncating, bit-shift process. If the data appears in color, it means the graphics mode will only allow 16 color variations, and the datasets are bit-shifted to 4-bit values. (Also, a 4-bit graphics mode will not allow help screen display or hardcopy EPS file creation.) Depending on the number of samples per trace in the recorded field data, the displayed data may show a greater or fewer number of samples. For example, a screen resolution of 1024 by 768 by 256 may be used to display a dataset with 1024 samples per trace, and also one with 256 samples per trace. In both cases, the data may occupy 512 of the available 768 vertical pixels. In the first case, only every other data sample is displayed. In the second case, every sample is repeated (displayed twice). Both result in the same size trace on the screen.

A time-variable gain is calculated and applied to data stored in a signed format only. (For 8-bit data, signed numbers vary from -127 to 128, unsigned numbers vary from 0 to 255. This came about because Sensor's & Software's data is signed and has no time-variable gain on it; GSSI's data is unsigned and already has gain applied.) The function simply finds the maximum amplitude in each of eight windows down the trace, and finds a gain in decibels that will make that amplitude 75% of the largest number available with the given bit-size. The time-variable gain is a decibel interpolation of the eight gain values. Every signed value is gained as it is read from disk to memory.

Only image displays are available; there is no wiggle trace option for the full screen display. The trace under the vertical cursor is always displayed in wiggle trace mode on the right side of the screen.

Caution: Pressing Shift-Enter on this display will exit the field display module and import the current cursor trace for comparison to the model response. Any previously imported trace being used for model response comparison will be lost. Importing a field trace can also change the thickness and permittivity of the first layer, and it can change the offset value. Pressing the Escape key will exit the field display module with no changes to the model.

If the field data held in memory is larger than what will fit on one screen, the Page Up and Page Down keys will scroll through the dataset one-half screen at a time. The Home and End keys will display the beginning or end of the data in memory, respectively. Pressing Control Right Arrow and Control Left Arrow will nudge the display forward or backward one trace at a time.

Within a given screen display of data, the vertical cursor (the vertical white line on the screen) can be moved one trace at a time with the Right and Left Arrow keys. As the cursor is moved, the trace it currently covers is updated on the right side of the screen as a wiggle plot. There is also a horizontal cursor (a horizontal white line on the screen) that can be moved up and down with the Up and Down Arrow keys. The sample number or two-way travel time it currently covers is displayed under the wiggle plot. The 'S' and 'T' keys are used to switch this displayed value. Both cursors can be jumped, 10 traces at a time for the vertical one and 10% of the total time for the horizontal one, by pressing Shift Arrow (Up, Down, Right, or Left). The zero time can be reset to wherever the horizontal cursor is at any time by pressing the '0' (zero) key.

Pressing the 'B' key will switch between a data display with and without the average of all the traces subtracted from every trace (background removal). As the data is read from disk, every trace is summed into a single set of trace values that are divided by the total number of traces read. This "average" trace is then subtracted from or added to every trace in the dataset when the 'B' key is pressed. The result is a removal of flat events that persist across the data. In cases of flat reflectors, this process will create a non-interpretable (useless) dataset. However, in cases where the system noise "rings" identically down every trace, and the targets are not flat and persistent across the data, this process will create a more interpretable dataset. Note that when Shift-Enter is pressed to import the cursor scan, the background removal process is not saved and the imported trace is always the data as it exists in the file on disk.

The F1 key will display a help screen summarizing the available active keys. The F2 key will scroll through various color palettes. Pressing the 'P' key will allow a hardcopy plot file of the current screen to be created. This is only possible with certain graphics modes, as described in a later section on hardcopy plot creation. If allowed, the resultant image file is in Encapsulated Post Script (EPS) format.

Three other keys are active at this point, and their functions are related. When the lateral spacing of the traces is known, the 'V' key will lead to a separate curve-fitting module that can be used to match hyperbolic curves to scattering from round targets. Matching the curves gives an estimate of the permittivity and, therefore, the velocity, such that depth can be approximated. Before the 'V' key will work, the program needs to know something about the lateral spacing between traces. This information can come from any of the following three methods.

First, when the data are in Sensor's & Software's .dt1 format and have been acquired at regularly spaced intervals, the trace spacing is read from the header and the 'V' key is active immediately. Second, when the data are in GSSI's SIR10A+ *.dzt* format and have been acquired at so many traces per second with marks taken at regularly spaced intervals, the marks are recognized by the program. When the 'V' key is pressed, the number of traces between each mark is counted. Traces are thrown out such that all mark intervals contain the same number of traces as was originally in the mark increment with the fewest traces. **Once this process has been done, the only way to return to a fully populated field data image is to Escape out and re-read the field file.** The image displayed after this process has been done is approximately spatially correct if the marks are accurate. If the marks are not accurate, for example, a mark is missing or an extra mark exists, the remaining two active keys, 'A' and 'R', can be used for adding or removing a mark, respectively. Move the vertical cursor to the appropriate trace and press the key for the action to take place at that trace.

The 'A' key is also the third way to input lateral trace spacing information. It can be used with data that does not fall into either of the above two categories. Move the cursor to any desired and known location and press 'A' to add a mark at that location. Move to any other known location and add another mark. As many marks as needed can be added, but a minimum of two are required. Once at least two marks exist, the 'V' key will be active between the marks.

Within the 'V' key curve-fitting module, a new set of active keys exists. Depending on the data type, two or three variable parameters are shown at the bottom of the screen. The average relative permittivity and the target radius are always shown. If marks are used to determine the lateral spacing, then the third variable parameter is the mark spacing in meters. The Left and Right Arrow keys move the highlight among the variable parameters. The highlighted parameter can be increased by pressing Home (+1.0), Up Arrow (+0.1), or Page Up (+0.01). It can be decreased by pressing End (-1.0), Down Arrow (-0.1), or Page Down (-0.01). A curve is drawn centered at the intersection of the vertical and horizontal cursors. This curve represents the arrival time of energy that scatters off the target and is received by the zero-offset antenna. The velocity between the antenna and the target is determined by the current value of the average relative permittivity according to

$$\text{Velocity} = \frac{c}{\sqrt{\varepsilon_r}},$$

where c is the speed of light in free space and ε_r is the average relative permittivity shown on the screen. Note that this simple formula assumes the relative magnetic permeability to be that of free space, or 1.0. The top of the curve can be moved by pressing the Control key with the Up, Down, Left, or Right Arrow keys. The Escape key exits the special 'V' key module. Depending on the graphics mode, the F1 key gives a help screen summary of active keys, and the 'P' key will open the option to make an EPS image file of the screen for hardcopy printing.

The two ways to exit the field data display module are the Escape key and the Shift-Enter key. Exiting with Escape closes the field file and returns to the Main Menu as if the field display module had never been entered. No modeling parameters are changed. This allows the field display to be used for looking at any datasets without affecting the current modeling effort. Exiting the field display with the Shift-Enter key initiates many tasks as described below.

Immediately after pressing Shift-Enter, the graphics screen is shutdown and all the field data in memory are released. The field file on disk is reaccessed, and the trace indicated by the cursor on exit is read into memory. Note that if the screen trace has had the average trace subtracted from it (background removed with the 'B' key), that same trace as read from the field data file will not have the background removed and will look different. If there is an error when trying to read the trace from the field data file, the program will return to the Main Menu as if Escape had been pressed.

After successfully reading the field trace, the program determines if a field trace has been introduced before to this set of model parameters. If it has not, or if it has but this field file name is different from the one most recently associated with this set of model parameters, then the field range-gain function and the current range-gain function are reset. The field range-gain is whatever time-variable gain was put on the data during acquisition according to the header information. The current range-gain is set to be the field range-gain plus the range-gain function calculated and applied to the displayed data if it is stored in a signed format. The field range-gain cannot be changed by the user while modeling. The current range-gain is always applied to the model response and the comparison field trace, and it can be changed any time by the user. See the next section for more information on the time-variable range-gain functions.

The number of samples in the model response is set to be the next power of two greater than or equal to the number of samples in the field trace (eg. 200 samples per field trace will set the model response to 256 samples). The sample interval time of the model response is set to be the total time of the field trace divided by the number of samples in the model response. This ensures that the model response will have the same total time as the field trace, but a number of samples that is a power of two. The starting transmission wavelet is rescaled such that its maximum amplitude matches the maximum amplitude of the ungained field trace. The field trace and the starting transmission wavelet are then resampled to the model output sample rate. The begin and end time are reset to zero and the total time, respectively; and the offset time is reset according to the zero time setting on the field data display.

If the first layer permittivity is still at the default setting of 1.00, it is reset to the final permittivity value from the 'V' key module (or the 'V' key default of 4.0 if 'V' was never pressed). If the permittivity is reset and the gain is reset and the first layer thickness is still at the default value of 1.0, then the first layer thickness is reset to the depth represented by the horizontal cursor at field view exit time (according to the velocity from the new first layer permittivity). The program variables associated with reading the field data from disk are also reset when a new field trace is imported.

Range (Time-Variable) Gain

Both the model response and an imported field trace can have time-variable gain (range-gain) applied. If an imported field trace exists, then the same time-variable gain function is always applied to both the model response and the field trace. The applied function is the current one shown by the Gain module on the Main Menu.

When the Gain module is entered from the Main Menu an interactive graphics screen is presented (figure 7). On the left is a linear plot of the current range-gain function. On the right is the same function plotted in decibels (log scale). The relationship between the two is according to the formula

$$dB = 20 \log10(\text{linear gain}),$$

so a multiplicative gain of 100 is equivalent to 40 decibels. Range-gain functions are defined on the decibel plot and shown on the linear plot for convenience.

Three range-gain functions are held in memory at all times. They are the current range-gain function, the field data range-gain (time-variable gain applied to the data during acquisition) and a simple two-point no gain function. Either of the last two can be selected immediately to be the current range-gain, in which case if the previous current range-gain was an independent function it is lost.

The Left and Right Arrow keys move the highlight among five choices as shown near the bottom of the screen. The two choices of Field Gain and No Gain are selected by pressing Enter when highlighted, and replace the current range-gain function. The other three choices are used to create a custom range-gain by editing the current function.

The number of nodes determines how many defined values of decibel range-gain will be given over the length of the trace. This number can vary from 2 to 8, and is changed by pressing the Up or Down Arrow key when this choice is highlighted.

To change the gain at a defined node number, first highlight the "Gain at Node _" choice. The Up and Down Arrow keys can now be used to select a node number. When the desired node is selected, press the Right Arrow to move the highlight to the gain value choice. This value, given in decibels, can now be increased or decreased. The Home (+3.0), Up Arrow (+1.0), and Page Up (+0.1) keys increase the value, and the End (-3.0),

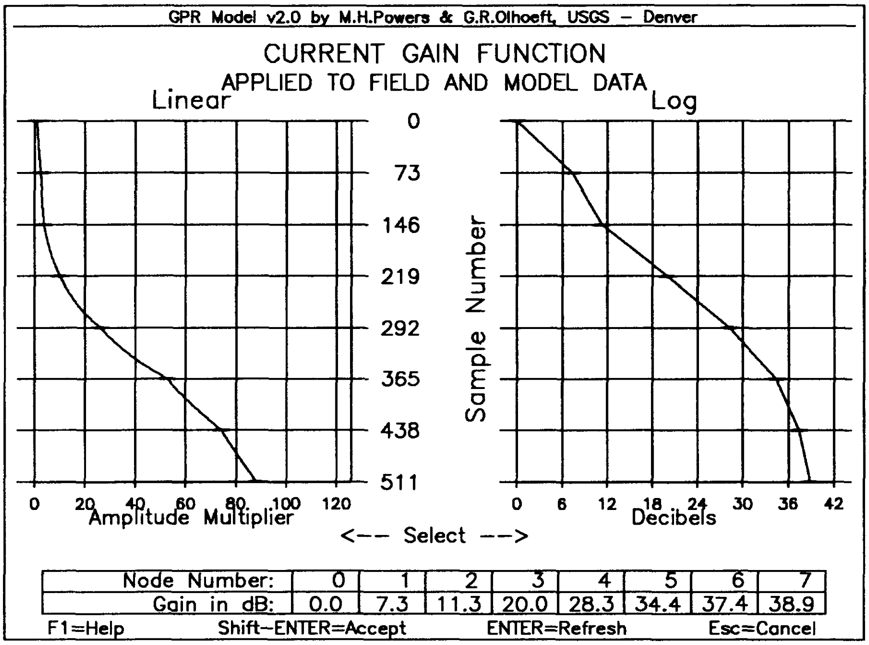

Figure 7. A hardcopy plot of the file created by pressing the 'P' key from the Gain module.

Down Arrow (-1.0), and Page Down (-0.1) keys decrease the value. When the desired value is shown, use the Left Arrow key to move the highlight back to the "Gain at Node _" choice to select another node for changing. Pressing Enter at any time will redraw the display to show the current values.

Two methods of exiting this module exist. Pressing Escape will exit without saving any changes. The previously current range-gain function shown upon entering the Gain module will remain in effect. Pressing Shift-Enter will exit and replace the previous range-gain function with the currently displayed one. The previous one is lost.

The range-gain function applied to the model response, and an imported field trace if it exists, is the linear function shown on the left side of the Gain module screen. This function is determined by linearly interpolating between the defined nodal values in decibels. The resulting fully defined decibel function is then converted to linear multiplicative values. This is definitely not the same as converting the decibel node values to linear multiplicative constants and interpolating between the constants.

When modeling without a field trace for comparison, the current range-gain, Field Gain, and No Gain are initially the same. The current range-gain can be set to anything and applied to the model response. When field data has been imported, the field range-gain and current range-gain are set as described in the previous section.

Model Building and Editing Utility

To add or delete layers to the model, the Model build/edit choice must be selected from the Main Menu. The resulting text window shows a spreadsheet of all the electromagnetic model parameters, any of which can be changed in this module. The highlight starts at the top of the screen on the number of layers. To add or delete layers, enter a number that is greater or less than the current number. A separate pop-up window will be created for entering the layer number(s) to add or delete. Added layers are created above the layer number specified, and inherit all of its parameters. When a layer number is given for deletion, that layer disappears and those below it are renumbered.

To move the highlight among the parameters, use the Tab and Shift Tab keys, the Page Up and Page Down keys, or the Up and Down Arrow keys. The Enter key will also move the highlight to the next parameter. The Right and Left Arrow keys only move the cursor within the digits of the highlighted parameter. Press F1 at any time for help information on the highlighted parameter. After changing a value and pressing Enter, it is important to check that the value shown is as desired. The values are entered as characters, interpreted as numbers, checked for validity, converted back to characters and re-displayed. This process can often result in something other than the value that was intended. Limits for each parameter are shown at the bottom of the screen in the Messages window.

Exiting this utility with the Escape key will lose all changes made. Control returns to the Main Menu and all parameter values are as they were before the utility was entered. To exit and keep all the parameter values as shown on the spreadsheet screen, press Shift-Enter.

Every parameter shown here can also be changed interactively while watching either the model response display or the frequency domain parameter display. The 'F2' utility within the frequency domain parameter display can also be used to determine the model parameters of a given soil mixture. See the Interactive Options section of this report for more detail.

Reading and Saving Program Files

While running this program to create stand-alone model responses or comparisons to field data, at any time the user can select Save from the Main Menu and store all key program variables. The intention is to be able to leave the program and return later to the same point as at exit time. The program files created by the Save utility are just ASCII text files containing the program parameter values in effect at the time of the save. In particular it is important to note that no field data traces or model responses are saved with the Save utility. Only the name of the path to the field data file and the required parameters for reading the right trace from the field file are stored. The model response is recreated from the stored model parameters. Four files are required to successfully return to a model with a field trace comparison: 1) the *gprmodv2.exe* executable file, 2) the appropriate starting wavelet file (eg. *ricker.trn*), 3) the saved program file (eg. *dunes01.gpm*), and 4) the field data file (eg. *\data\dunes.dat*).

An example program file is as follows:

```
USGS GPRMODEL Version 2.0 Parameter File
----------------------------------------
Model heading:  Yuma, Butler Site S0.5, Steel Pipe @ 37cm, 900 MHz
Starting wavelet pathname:  RICKER.TRN
Center frequency (MHz):  900
Number of layers in model:  2
Layer Parameters: εrl εr∞  τε  αε  μrl  μr∞  τμ  αμ  σ  thk
  5.50,  5.50,   0.0, 1.00,  1.00,  1.00,   0.0, 1.00,   30.00,  0.37
 -1.00, -1.00,   0.0, 1.00, 50.00, 50.00,   0.0, 1.00, 10000.00, infini
Number of samples per model trace:  512
Sample rate (ns):  0.048924
Begin time (ns):  0.000000
End time (ns):  20.890411
Offset (zero delay) time:  0.000000
Coupling ratio:  1.100000
Model only constant gain (dB):  9.000000
Display gain in dB:  0.0  3.0 36.0 45.0 46.0 47.0 48.0 49.0
Polarity flag:  1
Multiples on/off flag:  1
Reserved (currently t_sample*1000.):  69
Reserved (currently f_sample*1000.):  48
Reserved:
Reserved:
Reserved:
Field data flag:  1
Field file pathname:  \yuma\butler\J10F33NB.DZT
Trace number within field file:  120
Number of file header bytes:  1024
Number of samples per trace:  512
Bytes per extended trace:  1024
Bytes to first active sample:  0
Number of fixed bits per sample:  16
Total time per trace (ns):  25.000000
Field gain in dB:  10.1 10.1 42.4 47.2 57.9 57.9 64.5 66.1
First trace for field data screen display:  0
Sample number of cursor display:  123
```

These files can be stored with any acceptable DOS file name. Our convention has been to use the extension *.gpm* for convenience in viewing a listing of program files, but this is not required. Program files can be edited with any text editor. When the program reads the file, it uses the first line to decide whether or not to continue reading. The first line must be exactly as shown. Succeeding lines starting with /* or -- are ignored as comments. Of the succeeding lines that are not ignored, the order of the parameters is critical. For every parameter line, the characters preceding the colon are ignored. The parameter(s) is(are) taken to be the first non-blank character(s) after the colon. Parameter values are checked as they are read from the program file, but not with the same rigor with regard to compatibility as they are during a program run. For this reason, caution is required to avoid bombing the program by reading altered files.

Program files can be called two ways. First, on the command line at the DOS prompt the user can start the program and specify a previously saved program file at the same time by typing the program file name as a program argument (eg. *gprmodv2* *dunes01.gpm*). This bypasses the Main Menu and goes straight to the eXecute model screen. Second, the Read option can be chosen from the Main Menu. When the window asks for a filename, wildcard characters can be used to get a directory listing of all the appropriate files if desired. When a new program file is read, all previous parameter values are lost and control moves to the eXecute model screen. .

The only way to save work done while running the program is to use the Save option from the Main Menu. The program gives a prompt for verification before overwriting a previously existing file. It is a good idea to save not only after finalizing a model, but anytime something is to be redefined (such as a gain function or layer parameters). Saving a file has no effect on the current program variables; it just stores them. As a precaution, whenever the user exits the program naturally, a file called *lastmod.gpm* is saved with the parameter values at exit time. This file is overwritten every time the program is exited, so it is only meant to be a safeguard against accidental exits.

Hardcopy Plots

Every graphics screen has a utility for creating a plot file for a hardcopy image of that screen. Press the 'P' key to create a snapshot plot file. In the case of viewing images of field data sections, the 'P' key will only create an image file for three specific graphics modes. They are all 1024 by 768 by 256 color modes with specific graphics mode call numbers of 1024 (for Hercules Graphics Station GB1024 cards), 56 (for Tseng ET4000AX chip based cards), and 261 (for VESA (1991) standard mode). In this case the resulting file is in Encapsulated PostScript and can only be printed by a PostScript printer. Figure 8 is an example of such an image. The name of the EPS image file is automatically set to the same name as the field data file with a replaced extension of *.eps*. If a file with that name already exists, then it uses *gprdxxx.eps* where the *xxx* is an incremental number.

The other four graphics screens, the model response, the frequency dependence plot, the coupling ratio view of the starting wavelet, and the time-variable gain function, are stored as HPGL (Hewlett Packard Graphics Language) plot files (vector graphics). They can be viewed by many Windows programs and printed, through Windows, by any printer. Without Windows, these files can be printed by any device that is set up to recognize HPGL commands. The program will not overwrite an existing plot file; instead, the name is incremented. The user cannot select the name of any HPGL plot file prior to its creation, but the current name is always shown on the help screen. They are standardized to be *gprmxxx.plt*, *gprfxxx.plt*, *gprwxxx.plt*, and *gprgxxx.plt*, for the model response, the frequency dependence, the coupling ratio starting wavelet, and the range-gain screens, respectively; where the *xxx* is an incremental number. The file name can always be changed later with DOS commands. Figures 6 and 7 are examples of these plots from the Coupling ratio module and the Gain module, respectively. Examples of the plots created from the primary eXecute model screen and the frequency dependence screen are shown in figures 9 and 10. All of the figures in this paper were printed on a

Hewlett Packard Laserjet 4MP printer. Figure 8 was sent directly to the printer via the DOS copy command. The others, all HPGL files, were imported to Microsoft Word 5.0 for DOS and printed from there. The default plot mode creates HPGL files that only can be sent directly to an HPGL plotter (or imported to an interpreting program). By setting the DOS environment variable "SET LASERJET=TRUE", the HPGL plot files will be created to go directly (via the DOS copy command) to LaserJet compatible PCL (printer control language) devices.

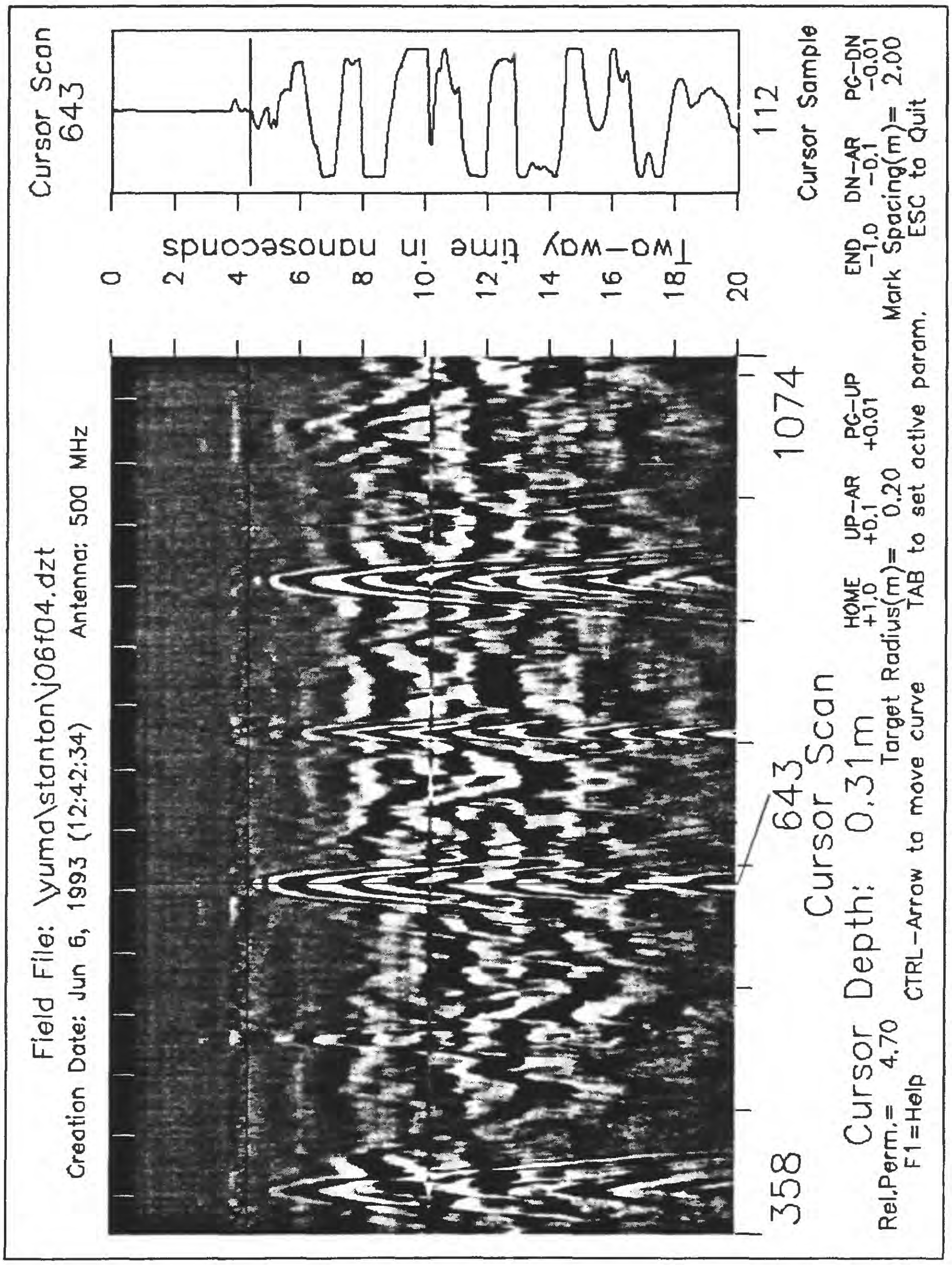

Figure 8. A hardcopy plot of the Encapsulated PostScript file created by making a screen capture in the field data viewing module. These files are only created when specific graphics modes are used. See the text for details.

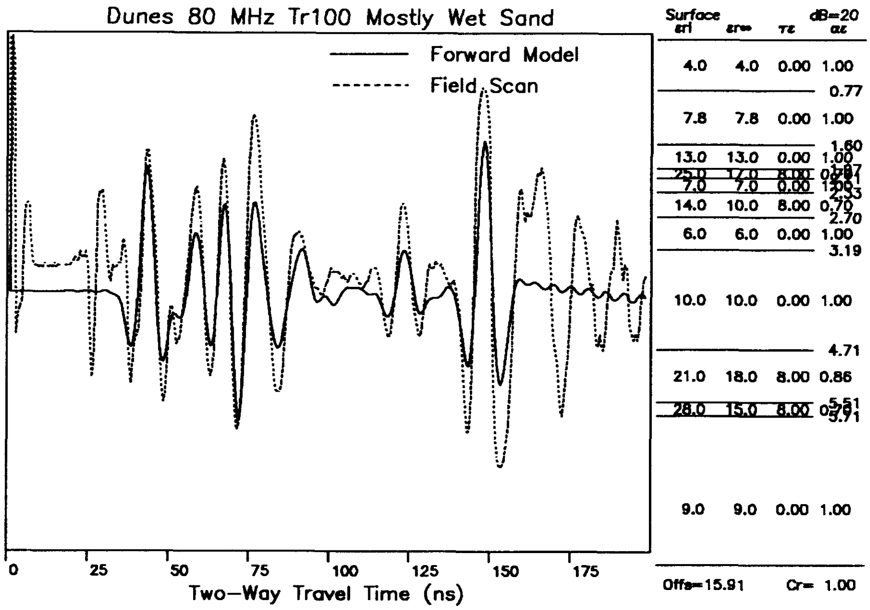

Figure 9. A hardcopy plot of the Hewlett Packard Graphics Language file created by using the 'P' key to perform a screen capture of the main model response screen.

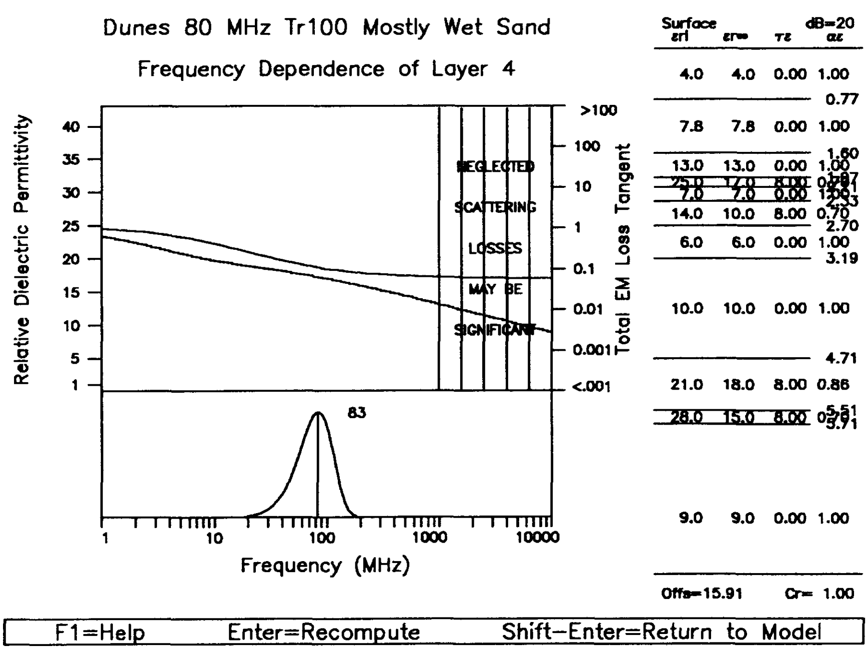

Figure 10. A hardcopy plot of the file created by using the 'P' key to perform a screen capture of the interactive frequency dependence screen.

Summary

This document is meant to be a user's manual and a technical explanation of a computer program designed as an aid in interpreting ground penetrating radar data. The only required input to the program is a supplied file containing a digitized starting wavelet. Information required for creating custom starting wavelets is provided. The program will optionally accept raw GPR field data for input and comparison to a model response. Previously saved program text files can also be used for input.

The primary output of the program is a screen display of a computer-generated GPR model response. The user can interactively vary the model parameters that describe the subsurface, while watching the effects on the model response. When field data has been imported, it is compared on the screen to the model response. Other program output includes a text file summary of the current model parameters, and optional plot files for creating hardcopy pictures of the screen. An example use of the program can be found in Sander (1994), where GPR data was used to examine contaminant flow through saturated sand. Some of the figures in her work were generated directly by this program.

This program is unique and valuable because it correctly accounts for the possibility of fully dispersive media. Any subsurface layer is allowed to have a velocity and attenuation rate that vary with frequency. The shape of the variations are controlled by a Cole-Cole distribution function that has been used to match the frequency dependencies measured on lab samples of soils (Olhoeft and Capron, 1994). The interactive nature of this program, and its availability on a personal computer platform, add to its usefulness as a learning tool.

Acknowledgments

Thanks are due to Kathy Sander and Jeff Lucius of the USGS, Branch of Geophysics, for their helpful criticisms as experimental users of the program. The reviews by Brian Rodriguez and Louise Pellerin of the USGS were also greatly appreciated.

References

Balanis, C.A., 1989, Advanced Engineering Electromagnetics: New York, John Wiley and Sons, 981p.

Bottcher, C.J.F., and Bordewijk, P., 1978, Theory of electric polarization: in Dielectrics in Time-Dependent Fields, Vol.2, Elsevier Publishing, Amsterdam, 561p.

Cole, K.S., and Cole, R.S., 1941, Dispersion and adsorption in dielectrics, I, alternating current characteristics: J. Chem. Phys., v.9, pp.341-351

Duke, S.K., 1990, Calibration of ground penetrating radar and calculation of attenuation and dielectric permittivity versus depth: MSc Thesis T-3920, Colorado School of Mines, Golden, CO, 236p.

GSSI, 1993, Appendix C - Radan file header: in RADAN instruction manual for RADAN 1, 2, & 3, Geophysical Survey Systems Inc., North Salem, NH, p1.

IEEE (The Inst. of Elec. and Electron. Engrs.), 1990, ANSI-IEEE Standard definitions of terms for radio wave propagation: IEEE Std. 211-1990, NY, IEEE, 24p.

Lorrain, D., and Corson, D.R., 1970, Electromagnetic fields and waves: San Francisco, Freeman, 706p.

May, B.T., and Hron, F., 1978, Synthetic seismic sections of typical petroleum traps: Geophysics, v.43, n.6, pp.1119-1147.

Olhoeft, G.R., and Strangway, D.W., 1974, Magnetic relaxation and the electromagnetic response parameter: Geophysics, v.39, n.3, pp.302-311.

Olhoeft, G.R., and Strangway, D.W., 1975, Electrical properties of the first 100 meters of the moon: Earth and Planetary Science Letters, v. 24, p. 394-404.

Olhoeft, G.R., 1981, Electrical properties of rocks: in Physical Properties of Rocks and Minerals, Y.S.Touloukian, W.R.Judd, and R.F.Roy, eds., McGraw-Hill, NY, pp.257-330.

Olhoeft, G.R., 1994a, libhpgl.lib 32-bit protected mode extended graphics library: The Austin Code Works, Austin, TX, version 9.3.

Olhoeft, G.R., 1994b, Modeling out-of-plane scattering effects: in Proceedings of the Fifth International Conference on Ground Penetrating Radar, June 12-16, Kitchener, Ontario, Canada, pp.133-144.

Olhoeft, G.R., and Capron, D.E., 1994, Petrophysical causes of electromagnetic dispersion: in Proceedings of the Fifth International Conference on Ground Penetrating Radar, June 12-16, Kitchener, Ontario, Canada, pp.145-152.

Pinson, J., 1991, Designing Screen Interfaces in C: Englewood Cliffs, NJ, Yourdon Press, 267p.

Powers, M.H., Duke, S.K., Huffman, A.C.III, and Olhoeft, G.R., 1992, GPRMODEL: One-dimensional full waveform forward modeling of ground penetrating radar data: U.S. Geological Survey Open File Report 92-532, 22p.

Sander, K.A., 1994, Characterization of DNAPL movement in saturated porous media using ground penetrating radar: ME Thesis ER-4336, Colorado School of Mines, Golden, CO, 258p.

Sen, P.N., Scala, C., and Cohen, M.H., 1981, A self-similar model for sedimentary rocks with application to the dielectric constant of fused glass beads: Geophysics, v.46, n.5, pp.781-795.

Sensors & Software, 1992, Appendix A - pulseEKKO data file format: in pE BASE - V4.0, Sensors & Software Inc., Mississauga, Ontario, p.A1-A2.

Sheriff, R.E., 1984, Encyclopedic dictionary of exploration geophysics, second edition: Society of Exploration Geophysicists, Tulsa, OK, 323p.

Stratton, J.A., 1940, Electromagnetic theory: McGraw-Hill, NY, 615p.

VESA (Video Electronics Standards Association), 1991, Super VGA BIOS Extension, VESA Super VGA Standard VS911022: 20 October 1991, San Jose, VESA, 25p.

Ward, S.H., and Hohmann, G.W., 1988, Electromagnetic Theory for Geophysical Applications, in Electromagnetic Methods in Applied Geophysics: Volume 1, Theory, M.N.Nabighian ed., Society of Exploration Geophysics, Tulsa, OK, pp. 130-311.

Appendix A: GSSI SIR10A+ Data Format

The following statements are from the C include file that is used by the program to read the *.dzt* data format. They are presented here as an explanation of the data format. This code was written by Jeff Lucius of the Branch of Geophysics in the U.S. Geological Survey. This format is used by GSSI's RADAN software package and is published in the RADAN manual (GSSI, 1993).

```
/* Make sure the following structures are packed on byte boundaries! */
#if defined(_INTELC32_)
 #pragma align (DztDateStruct=1)
 #pragma align (DztHdrStruct=1)
 #pragma align (DztHdrStructAlt=1)
#elif defined(__BORLANDC__)
 #pragma option -a-
#elif defined(_MSC_VER)
 #pragma pack(1)
#elif defined(__ZTC__)
 #pragma ZTC align 1
#endif
struct DztDateStruct
{  unsigned sec2  : 5;    /* second/2  (0-29) */
  unsigned min   : 6;    /* minute    (0-59) */
  unsigned hour  : 5;    /* hour      (0-23) */
  unsigned day   : 5;    /* day       (1-31) */
  unsigned month : 4;    /* month     (1-12; 1=Jan, 2=Feb, etc. */
  unsigned year  : 7;    /* year-1980 (0-127; 1980-2107) */
} ; /* 4 bytes (32 bits) if tightly packed */

struct DztHdrStruct
{  unsigned short rh_tag;        /* 0x00ff if header */
  unsigned short rh_data;        /* offset to data in file or could be
              the size of the header structure */
  unsigned short rh_nsamp;       /* samples per scan */
  unsigned short rh_bits;        /* bits per data word */
  unsigned short rh_zero;        /* binary offset (mid-point of data) */
  float       rh_sps;         /* scans per second */
  float       rh_spm;         /* scans per meter */
  float       rh_mpm;         /* meters per mark */
  float       rh_position;    /* position (ns) */
  float       rh_range;       /* range (ns) */
  unsigned short rh_npass;       /* scans per pass for 2D files */
  struct DztDateStruct rh_create; /* date created */
  struct DztDateStruct rh_modif; /* date modified */
  unsigned short rh_rgain;       /* offset to range-gain function */
  unsigned short rh_nrgain;      /* size of range-gain function */
  unsigned short rh_text;        /* offset to text */
  unsigned short rh_ntext;       /* size of text */
   unsigned short rh_proc;       /* offset to processing history */
  unsigned short rh_nproc;       /* size of processing history */
   unsigned short rh_nchan;      /* number of channels */
   float       rh_epsr;       /* average dielectric constant */
   float       rh_top;        /* top position in meters */
   float       rh_depth;      /* range in meters */
   char        reserved[32];
   char        rh_antname[14]; /* antenna name */
   unsigned short rh_chanmask;   /* active channels mask */
  char       rh_name[12];    /* this file name */
  unsigned short rh_chksum;      /* checksum for header */
   /* 128 bytes to here */
   char        variable[896];  /* range-gain, comments, and processing
          history */
} ; /* 1024 bytes if tightly packed */
/*
```

```
 * WARNING! the rh_data field may not actually be the offset to data in
 *       multi-channel files! It is probably best to read in all the
 *       headers which leaves the file pointer at the start of the first
 *       scan (these programs do that) or read in the first header to
 *       discover the number of channels than rewind the file and move
 *       the file pointer "number of channels times sizeof(the header
 *       structure)".
 */
/* Here is an equivalent structure with better field names */
struct DztHdrStructAlt
{  unsigned short dzt_hdr_id;
   unsigned short offset_to_data;
   unsigned short samples_per_scan;
  unsigned short bits_per_sample;
  unsigned short amp_midpoint;
   float       scans_per_second;
   float       scans_per_meter;
   float       meters_per_mark;
   float       start_time_offset;
   float       total_time_range;
   unsigned short scans_per_pass;
   struct DztDateStruct create_date;
   struct DztDateStruct modified_date;
  unsigned short offset_to_rgain;
  unsigned short sizeof_rgain;
  unsigned short offset_to_text;
  unsigned short sizeof_text;
  unsigned short offset_proc_hist;
  unsigned short sizeof_proc_hist;
  unsigned short number_of_channels;
  float       ave_rel_diel_perm;
  float       top_in_meters;
  float       range_in_meters;
  char        reserved[32];
  char        ant_number[14];
  unsigned short channel_mask;
  char        this_filename[12];
  unsigned short checksum;
  /* 128 bytes to here */
  char        variable[896];
} ; /* 1024 bytes if tightly packed */
```

The field range-gain is further encoded within the above format. It is interpreted through the following code which is in the subroutine import_field in the program *gprmodv2.c*.

```
*********** get range-gain *****************************/
        nrgbp=*(unsigned short *)((char *)hdrval[num]+hdrval[num]->rh_rgain);
        lfnumrgbp=(int)nrgbp;
        for (i=0;i<(lfnumrgbp);i++)
        {
                lfgainpts[i]=*(float *)((char *)hdrval[num]+hdrval[num]->rh_rgain+2+(4*i));
        }
```

The parameter nrgbp is the number of range-gain breakpoints (number of defined decibel gain values over the total time of the trace), and the array lfgainpts holds the values in decibels at the breakpoints. These decibel values are linearly interpolated between each breakpoint to get the decibel gain for every sample in the trace.

Appendix B: Sensors & Software's pulseEKKO Data Format

For every field data file in this format, there is a separate associated header file. The data files are always given a filename extension of *.dt1*, and the associated header file has the same filename with the extension *.hd*. The *gprmodv2* program always expects the *.dt1* filename to be given as the field data filename. Every trace in the *.dt1* data file has a trace header of 25 4-byte floating point values and 28 1-byte characters. The program reads the trace header of the first trace in the *.dt1* data file, and then reads the associated header file before reading the rest of the data. An example header file and our code to read this format are listed here.

This is the example header file.

```
8000
pulseEKKO IV radar survey over tanks
09/10/88
NUMBER OF TRACES  = 121
NUMBER OF PTS/TRC  = 118
TIMEZERO AT POINT  = 18
TOTAL TIME WINDOW  = 94
STARTING POSITION  = 1.000000
FINAL POSITION     = 61.000000
STEP SIZE USED     = 0.500000
POSITION UNITS     = m.
NOMINAL FREQUENCY  = 200.000000
ANTENNA SEPARATION = 1.000000
PULSER VOLTAGE (V) = 400
NUMBER OF STACKS  = 256
SURVEY MODE       = Reflection
SIGNAL SATURATION CORRECTION APPLIED
THIS FILE A MERGING OF TANKS1.dt1 AND TANKS2.dt1.
ELEVATION DATA ENTERED : MAX = 15.000000  MIN = 12.000000
```

This is the subroutine from *gprmodv2.c* used to read *.dt1* data format.

```
/***************************************************************************/
int GetSSHdrs(int *nscans,int *nsamps,int *pt0,double *ttime,double *x0,
                                        double *xe,double *xi,double *ssfreq,double *antsep,
                                        char *dflnm,FILE *dfile,char *hddate,char *comments)
{
        float tracehdr[25];
        int i,j,pcnt,paramlinecnt=0,dateline=10,lpline=20;
        char hdfilename[MAX_PATHLEN], *s, tmp[160];
        char first_two[3], *p[50], paramfileline[80];
        FILE *hd_file;

        rewind(dfile);
        /* reads first trace header (25 values) in .dt1 data file */
        if (fread((void *)tracehdr,25*sizeof(float),1,dfile)<1) return 1;
        if (tracehdr[5]!=2) return 1;

        s=dflnm; hdfilename[0]=0;
        while ((*s)!='.') { strncat(hdfilename,s,1); s++; }
        strncat(hdfilename,".hd",3);

        if ((hd_file = fopen(hdfilename,"rt")) == NULL)
        {
                printf("\npulseEKKO format should have separate header file.\n");
                printf("Unable to open %s.\n",hdfilename);
                return 1;
        }

        while ((fgets(paramfileline,79,hd_file))!=NULL)
```

```
	{
		strncpy(first_two,paramfileline,2);
		if ( (!strncmp(first_two,"\n\0",2)) || (!strncmp(first_two," \n",2)) \
		|| (!strncmp(first_two,"  ",2)) ||   (!strncmp(first_two,"/*",2)) ) continue;

		paramlinecnt++;
		if (paramlinecnt==1) continue;

		if (paramlinecnt<dateline)
		{
			if (paramfileline[2]=='/' && paramfileline[5]=='/')
				{ strncpy(hddate,paramfileline,8); hddate[8]=0; dateline=paramlinecnt; }
			else strcat(comments,paramfileline);
			continue;
		}

		pcnt=brkstr(p, paramfileline, SETA);
		if (!strncmp(p[0],"NUMBER OF TRACES",16)) { *nscans=atoi(p[1]); continue; }
		if (!strncmp(p[0],"NUMBER OF PTS/TRC",17)) { *nsamps=atoi(p[1]); continue; }
		if (!strncmp(p[0],"TIMEZERO AT POINT",17)) { *pt0=atoi(p[1]); continue; }
		if (!strncmp(p[0],"TOTAL TIME WINDOW",17)) { *ttime=atof(p[1]); continue; }
		if (!strncmp(p[0],"STARTING POSITION",17)) { *x0=atof(p[1]); continue; }
		if (!strncmp(p[0],"FINAL POSITION",14)) { *xe=atof(p[1]); continue; }
		if (!strncmp(p[0],"STEP SIZE USED",14)) { *xi=atof(p[1]); continue; }
		if (!strncmp(p[0],"NOMINAL FREQUENCY",17)) { *ssfreq=atof(p[1]); continue; }
		if (!strncmp(p[0],"ANTENNA SEPARATION",18)) { *antsep=atof(p[1]); continue; }

		strcat(comments,p[0]); i=1;
		while (pcnt>i) { strcat(comments,"="); strcat(comments,p[i]); i++; }
		strcat(comments,"\n");
	}

	if (tracehdr[2]!=(float)(*nsamps)) return 1;
	if (tracehdr[8]!=(float)(*ttime)) return 1;

	fclose(hd_file); rewind(dfile);
	return 0;
}
```

The 25 trace header values in the *.dt1* file are detailed in the Appendix of the pulseEKKO manual (Sensors & Software, 1992).

Appendix C: Data Format Conversion Utility

Three utility programs are included with this GPR modeling package. They were specifically written to convert data acquired by a GSSI SIR7 system and recorded with a DT6000 tape unit, into the above described SIR10A+ *.dzt* format. The data recorded by the DT6000 consist of 512 8-bit values per trace. They are recorded with no file header and no trace headers. These utility programs can be used to convert any fixed-bit data traces into the *.dzt* format if all of the original trace and file headers are first stripped off.

The range-gain used in the field with the SIR7 is recorded by saving reference traces that consist of an oscillating signal sent through the Time Gain Amplifier. The amplitude envelope of the oscillations gives a measure of the time-relative gain. The total time (range) represented by the 512 samples of each trace (and, therefore, the sample interval time) is determined by recording reference traces that consist of an oscillating signal where the period of each oscillation is 10 nanoseconds.

In the *.dzt* format, the range-gain is stored in the file header as two to eight breakpoint values in decibels. The first breakpoint always represents the gain of the first sample in the trace. The last breakpoint always represents the gain of the last sample in the trace. When more than two breakpoints are used, the other breakpoints are equally spaced between the first one and the last one.

The utility program called *gainpts.exe* reads the SIR7 generated oscillating range-gain traces, and estimates the gain by interpolating the amplitude envelope. Figure A1 shows an example of the first of two graphics screens displayed by *gainpts*. Eight range-gain traces have been plotted on top of each other and their amplitude envelope has been determined. Because these traces are always 8-bit values, the amplitudes vary from -127 to 128. A maximum amplitude of 128 is considered to be 100% gain and is arbitrarily set equal to 40 decibels. The rest of the amplitude envelope is then converted to decibels, and digitized at 8 breakpoints. The second screen (figure A2) shows the original envelope function in decibels, and the digitized and linearly interpolated one on the same decibel scale. It also shows the original linear amplitude envelope function in percent, where the maximum recordable amplitude is 100%, and the equivalent interpolated decibel function converted back to percent on the same scale. The purpose of this program is to get the eight decibel breakpoint values that will be put into the header of the new .dzt file. These values are shown on the *gainpts* screen and can optionally be saved in a text file report.

In the *.dzt* format, the total two-way traveltime (listen time or range) represented by each trace is stored in the file header as a single floating point value in nanoseconds. The utility program called *timecal.exe* reads the SIR7 generated oscillating time calibration traces to determine the number of samples in each 10 nanosecond period. Three methods of estimating the number of samples per period are displayed on a single graphics screen (figure A3), along with a statistical summary of the resulting measures of the total time represented by the trace. The user must select the value of total time believed to be appropriate given the information presented. An optional text file summarizing the statistical picks can also be generated at this point.

Once the total time (range) in nanoseconds and the range-gain function in decibels are known, a parameter file that summarizes the necessary file information is created. The file can be named anything and is the input to the third utility program called *mksir10.exe*. This program converts the information in the parameter file into a *.dzt* format file header, and appends it to the original data. The resulting file can be read directly by *gprmodv2*. An example of the *mksir10* parameter file is listed below.

1) Data filename: data\dunes80.dat
2) Total time (range): 301.2
3) Gain (2-8 values): 0.0 7.3 11.3 20.0 28.3 34.4 37.4 38.9
4) Samples per scan: 512
5) Bits per sample: 8
6) Scans per second: 51.2
7) Scans per meter: 0
8) Meters per mark: 10
9) Position: 0.0
10) Antenna: 3112
11) Comments: Great Sand Dunes 80 MHz data from dune onto frozen Medano creek - from dunes88\dunes8_1.dat 9500-10500

More information about the parameter file, and a copy of one, can be obtained by executing the *mksir10* program without specifying any input filename (just type *mksir10* and return). Information about each of the three utility programs is also available in the comments at the top of the source code provided in the files *gainpts.c*, *timecal.c*, and *mksir10.c*.

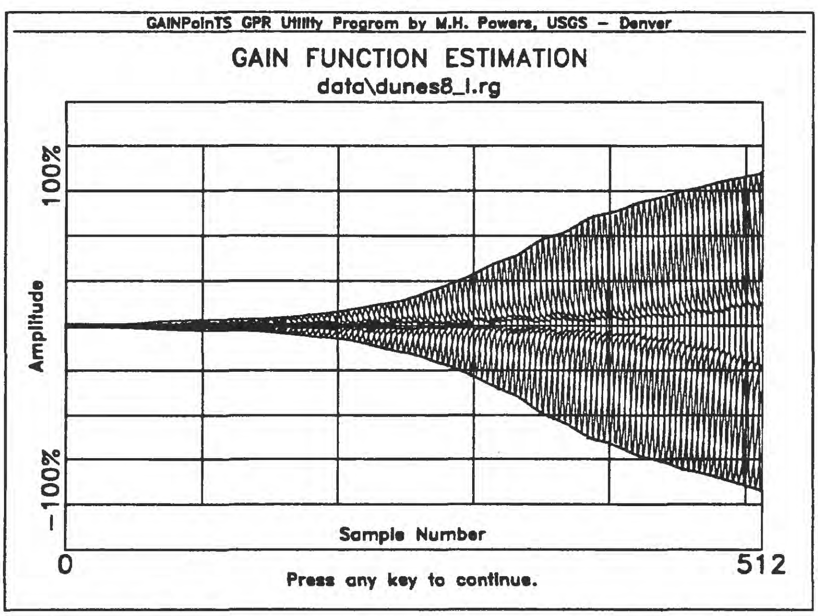

Figure A1. The first of two graphics screens displayed by the utility program *gainpts*. This screen shows the interpolated (and extrapolated if necessary) amplitude envelope function created after superimposing eight oscillating range-gain traces created by older GSSI systems.

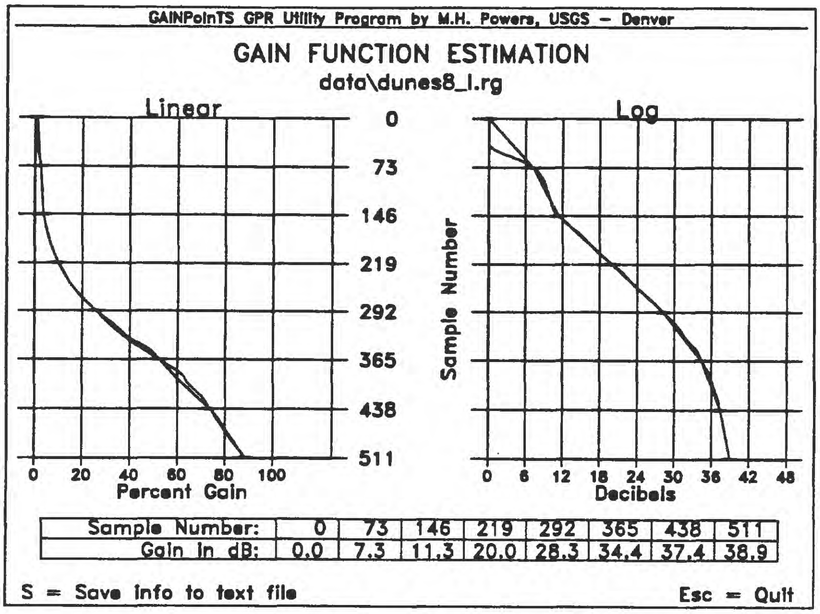

Figure A2. The second and final graphics screen displayed by the utility program *gainpts*. The amplitude envelope has been sampled at eight "node points" down the trace. The gain function approximated by using a linear interpolation of the decibel values between the eight points is shown on both scales, along with the original, fully populated function.

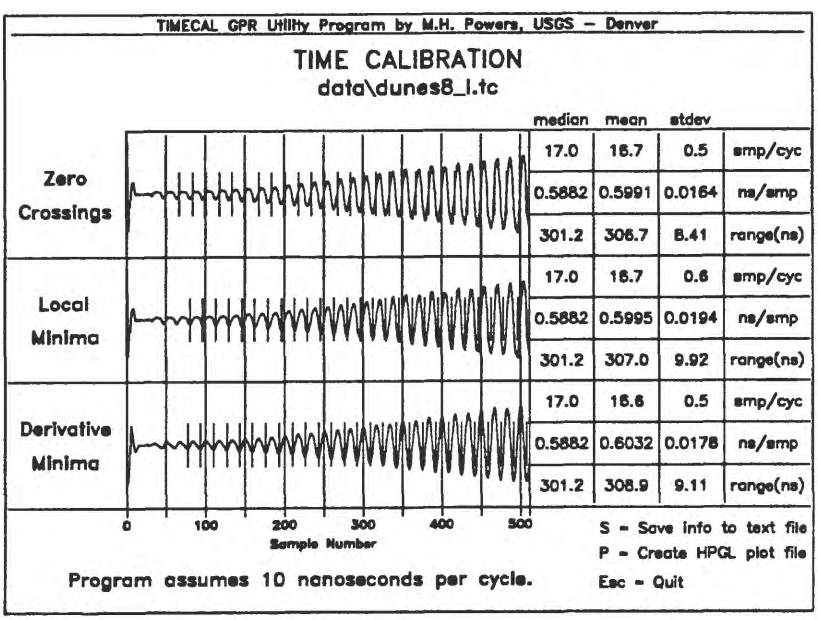

Figure A3. The graphics screen displayed by the utility program timecal. The purpose of this program is to determine the total time (range) of the traces recorded along with this particular time-calibration trace. The time-calibration traces have oscillations with periods of 10 nanoseconds. The user must select the total time based on this display.